

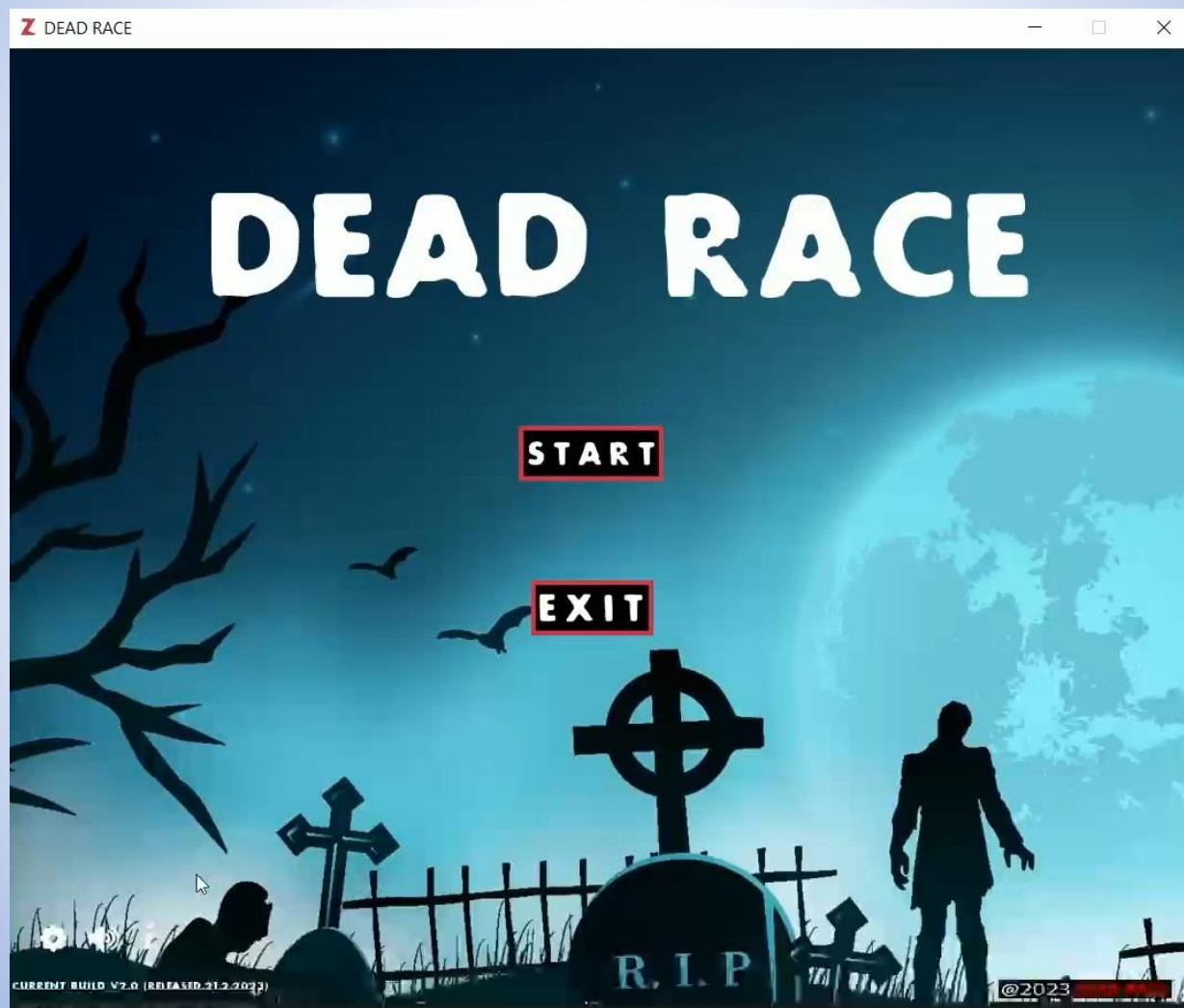
Dead Race

The game of survival





DEAD RACE

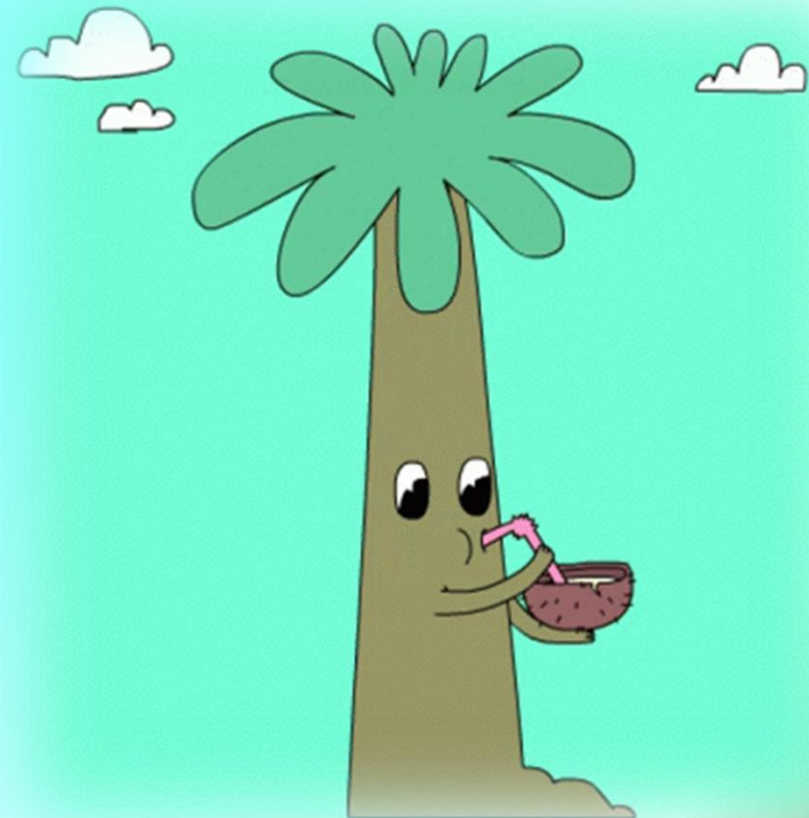


Made by Khushal (2023)



Index

- Objective
- Introduction
- Software and hardware requirements
- Flow chart
- Conclusion and future scope
- Result
- Vote of thanks





Objective



- ❑ The game is developed for full time entertainment and enthusiasms. It teaches the gamer to be alert at every situation. Though the purposed game is an action game, it doesn't involve direct involve. No animal killing or human killing is performed in the game,so it can also be viewed as a non-violence game.
- ❑ Kids can also play this game, because the design of the game is very controlling the game is very easy-pressing some neighboring keys of the keyboard



Introduction



Dead race is a 2D game which is completely based on python language. It has all its coding in python language. It is a game with different levels which get tough by each passing of it. In this game the character has to fight with zombies and has to cross the different huddles. Unlike most other video games genres which are classified by their game play this game is based on narrative and visual presentation and uses a variety of gameplay types. It uses certain modules as pygame for graphics. Though the proposed game is an action game but it doesn't involve direct violence. Here character has to survive the deadliest with different weapons like guns and grenade. It teaches the gamer to be alert at every situation they faces. Goals are fundamental to games; they determine what player has to do to win the game, and give the player a sense of accomplishment and progression.

THE WAR AGAINST THE DEAD HAS JUST BEGUN!



Software and hardware requirements

System requirements

- Operating system- 7,8,10,11, XP
- Cpu- 2.4 Ghz, Pentium 4
- Gpu- 128 mb , video card
- Memory- 512 mb ram
- Free space- 320 mb
- Software- python, code editor,

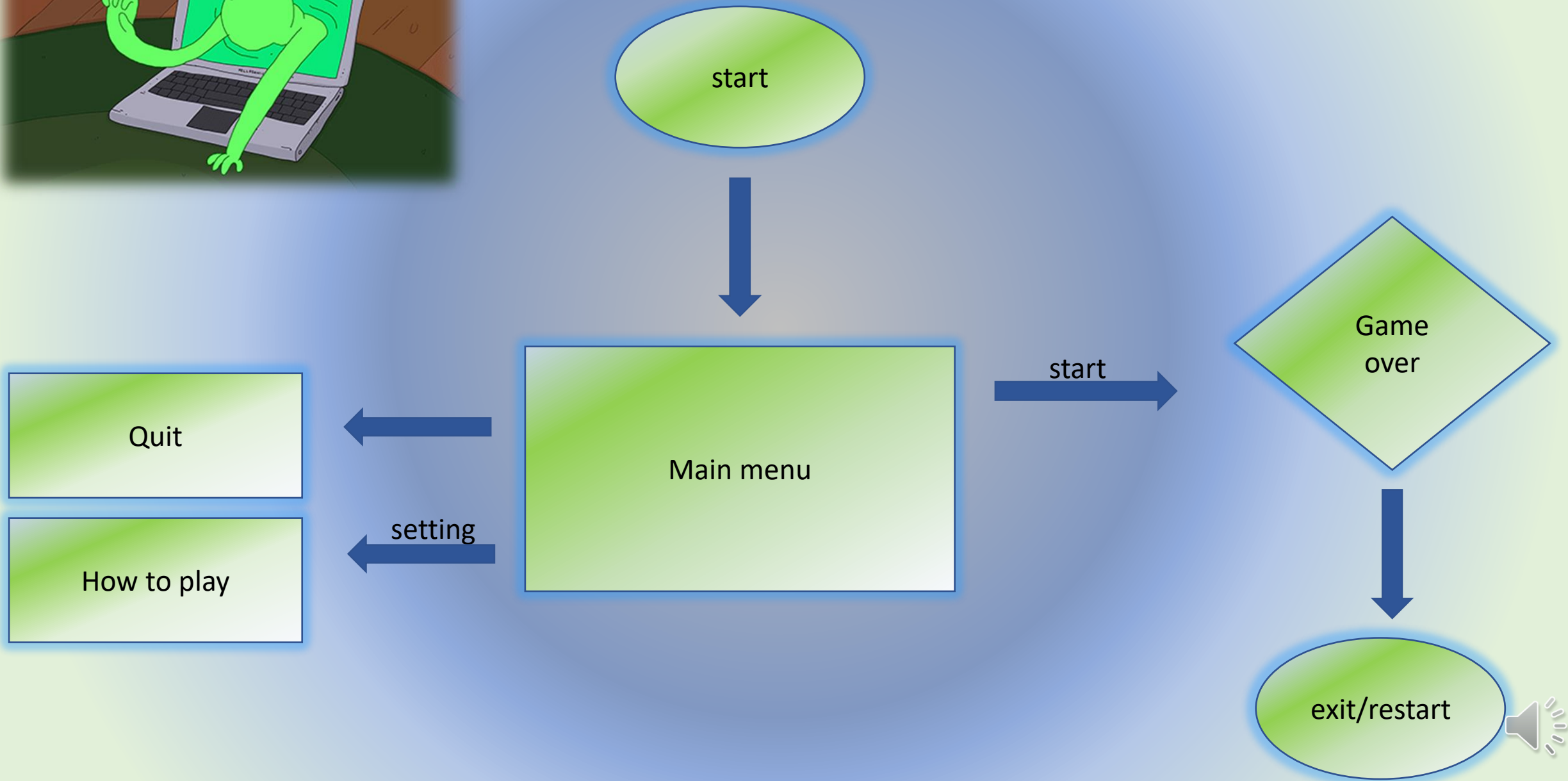
Requirement packages

Hardware requirements

- Keyboard
- Mouse
- Headset
- Microphone
- Computer
- Monitor



Flow chart



Conclusion And Future Scope

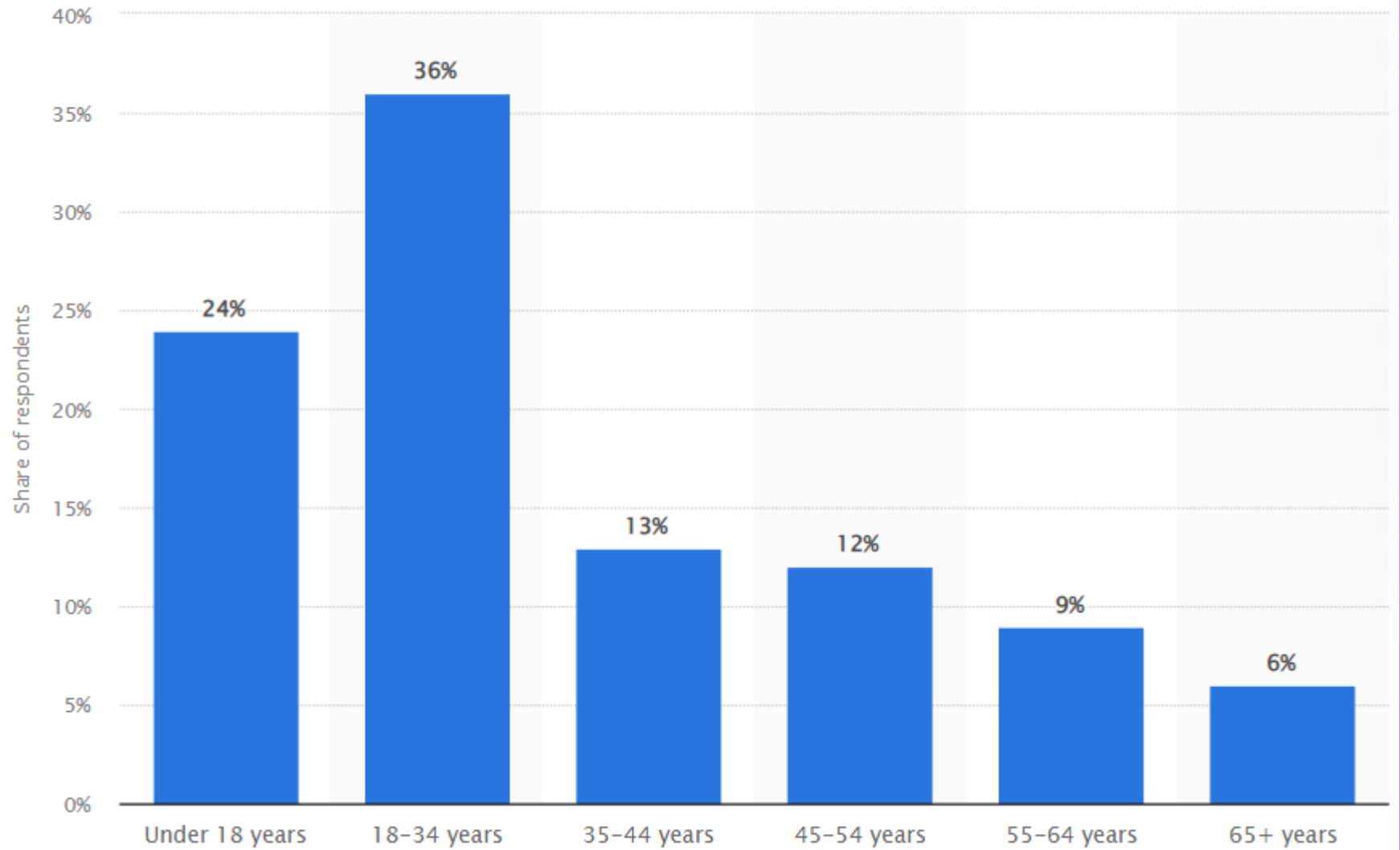


- ☐ The whole purpose of the project is to start students in thinking about survival techniques and ideas.
- ☐ It test the limits of your capabilities by shedding self doubt and fear of failure, but be realistic.
- ☐ This project is used to provide a virtual image for the combination of both structured and unstructured information of this project “Dead Race”.
- ☐ This is a single player strategy game. It has an attractive background, different enemy type, graphics perfect for horror games, live score and background music.





Distribution of
video gamers
in 2023 by age
group as
followed:-



```
MAINGAME.py X game1.py game2.py
MAINGAME.py > ...
1 import pygame
2 import os
3 import random
4 import csv
5 import button
6 import pygameui
7 import webbrowser
8 import time
9 import itertools
10
11 pygame.init()
12
13 FULL_SCREEN_WIDTH = 1080
14 FULL_SCREEN_HEIGHT = 1920
15 SCREEN_WIDTH = 1080
16 SCREEN_HEIGHT = int(SCREEN_WIDTH * 0.8)
17 programIcon = pygame.image.load('img/favicon/favicon-main.png')
18 screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
19 pygame.display.set_icon(programIcon)
20 pygame.display.set_caption('DEAD RACE')
21 pygame.mouse.set_visible(1)
22
23 # FPS
24 clock = pygame.time.Clock()
25 FPS = 85
26 PING = [4, 5]
27
28
29 # VAR
30 GRAVITY = 0.50
31 SCROLL_THRESH = 200
32 ROWS = 16
33 COLS = 150
34 TILE_SIZE = SCREEN_HEIGHT // ROWS
35 TILE_TYPES = 22
36 MAX_LEVELS = 6
37 screen_scroll = 0
38 bg_scroll = 0
39 level = 1
40 start_game = False
41
42
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
main/Zombioo-main/MAINGAME.py
pygame 2.1.3 (SDL 2.0.22, Python 3.9.13)
Hello from the pygame community. https://www.pygame.org/contribute.html
PS C:\Users\hp\Desktop\Zombioo-main\Zombioo-main>
```

First we have imported pygame, then we have added FPS-85, then we have put the gravity for the player to jump. Then we start the game.

```
MAINGAME.py X game1.py game2.py
MAINGAME.py > ...
62
63 # IMAGES
64 start_img = pygame.image.load('img/btn/start_btn.png').convert_alpha()
65 exit_img = pygame.image.load('img/btn/exit_btn.png').convert_alpha()
66 restart_img = pygame.image.load('img/btn/restart_btn.png').convert_alpha()
67 settings_img = pygame.image.load('img/btn/cog.png').convert_alpha()
68 jk_img = pygame.image.load('img/btn/jk.PNG').convert_alpha()
69 menubg = pygame.image.load('img/background/Graveyard.jpeg').convert_alpha()
70 speaker_img = pygame.image.load('img/btn/speaker.png').convert_alpha()
71 speaker_muted_img = pygame.image.load('img/btn/speaker_muted.png').convert_alpha()
72 info_img = pygame.image.load('img/btn/info_button.png').convert_alpha()
73 angel_statue = pygame.image.load('img/tile/20.png').convert_alpha()
74
75 #KEYBOARD
76 Wkey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/W_Key_Light.png').convert_alpha()
77 Akey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/A_Key_Light.png').convert_alpha()
78 Dkey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/D_Key_Light.png').convert_alpha()
79 Qkey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/Q_Key_Light.png').convert_alpha()
80 ESCkey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/Esc_Key_Light.png').convert_alpha()
81 SPkey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/Spacelarge_Key_Light.png').convert_alpha()
82 Mkey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/M_Key_Light.png').convert_alpha()
83 Ukey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/U_Key_Light.png').convert_alpha()
84 Fkey = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/F_Key_Light.png').convert_alpha()
85 F5key = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/F5_Key_Light.png').convert_alpha()
86 F4key = pygame.image.load('img/icons/keyboard/Keyboard & Mouse/Light/F4_Key_Light.png').convert_alpha()
87
88 # BACKGROUND IMAGES
89 pine1_img = pygame.image.load(
90     'img/background/2_background_NEST/pxfuel.jpg').convert_alpha()
91 pine2_img = pygame.image.load(
92     'img/background/2_background_NEST/pxfuel.jpg').convert_alpha()
93 mountain_img = pygame.image.load('img/background/2_background_NEST/pxfuel.jpg').convert_alpha()
94 sky_img = pygame.image.load('img/Background/sky_cloud.png').convert_alpha()
95 headhp = pygame.image.load('img/player/headHP.png').convert_alpha()
96 headdeadhp = pygame.image.load('img/player/headdeadHP.png').convert_alpha()
97 border_settings_img = pygame.image.load('img/background/border_settings.png').convert_alpha()
98 border_lore_img = pygame.image.load('img/background/border_lore.png').convert_alpha()
99
100
101
102 # TILES ETC
103
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
main/Zombioo-main/MAINGAME.py
pygame 2.1.3 (SDL 2.0.22, Python 3.9.13)
Hello from the pygame community. https://www.pygame.org/contribute.html
PS C:\Users\hp\Desktop\Zombioo-main\Zombioo-main>
```

In this we have inserted images for each level and we have used key A,D and W to move forward ,backward and to jump respectively.




```
Run Terminal Help
MAINGAME.py - Zombioo-main - Visual Studio Code

MAINGAME.py x game5.py game1.py game2.py
MAINGAME.py > ...

88 # BACKGROUND IMAGES
89 pine1_img = pygame.image.load(
90     'img/background/2_background_NEST/pxfuel.jpg').convert_alpha()
91 pine2_img = pygame.image.load(
92     'img/background/2_background_NEST/pxfuel.jpg').convert_alpha()
93 mountain_img = pygame.image.load('img/background/2_background_NEST/pxfuel.jpg').convert_alpha()
94 sky_img = pygame.image.load('img/Background/sky_cloud.png').convert_alpha()
95 headhp = pygame.image.load('img/player/headHP.png').convert_alpha()
96 headdeadhp = pygame.image.load('img/player/headdeadHP.png').convert_alpha()
97 border_settings_img = pygame.image.load('img/background/border_settings.png').convert_alpha()
98 border_lore_img = pygame.image.load('img/background/border_lore.png').convert_alpha()
99
100
101
102 # TILES ETC
103 img_list = []
104 for x in range(TILE_TYPES):
105     img = pygame.image.load(f'img/Tile/{x}.png')
106     img = pygame.transform.scale(img, (TILE_SIZE, TILE_SIZE))
107     img_list.append(img)
108
109 bullet_img = pygame.image.load('img/icons/ammo.png').convert_alpha()
110 bullet_zombie = pygame.image.load('img/icons/bulletzombie.png').convert_alpha()
111
112 grenade_img = pygame.image.load('img/icons/grenade_box.png').convert_alpha()
113 molotov_img = pygame.image.load('img/icons/molotov.png').convert_alpha()
114
115 health_box_img = pygame.image.load('img/icons/health_box.png').convert_alpha()
116 ammo_box_img = pygame.image.load('img/icons/ammo_box.png').convert_alpha()
117 grenade_box_img = pygame.image.load('img/icons/grenade_box.png').convert_alpha()
118 molotov_box_img = pygame.image.load('img/icons/molotov_box.png').convert_alpha()
119 item_boxes = {
120     'Health' : health_box_img,
121     'Ammo' : ammo_box_img,
122     'Grenade' : grenade_box_img,
123     'Molotov' : molotov_box_img
124 }
125
126
127 # COLORS
128 BG = (81, 6, 13)
```

After inserting background images we have placed tiles on which player runs and escape from zombies .

```
MAINGAME.py x game5.py game1.py game2.py
MAINGAME.py > ...

126
127 # COLORS
128 BG = (81, 6, 13)
129 RED = (176, 8, 12)
130 WHITE = (255, 255, 255)
131 GREEN = (26, 110, 15)
132 BLACK = (0, 0, 0)
133 CRIMSON = (191, 46, 72)
134
135 #MUSIC
136 menumusic = pygame.mixer.music.load('audio/THEME.wav')
137 pygame.mixer.music.play(0)
138 SHOOT_SOUND = pygame.mixer.Sound('audio/shot.mp3')
139 SHOOT_SOUND.set_volume(0.6)
140
141 RELOAD = pygame.mixer.Sound('audio/Reloading.mp3')
142 RELOAD.set_volume(1)
143
144 GRENADESOUND = pygame.mixer.Sound('audio/grenade.mp3')
145 GRENADESOUND.set_volume(1)
146
147 MOLOTOVSOUND = pygame.mixer.Sound('audio/molotov.wav')
148 MOLOTOVSOUND.set_volume(1)
149
150 MOLOTOVBR = pygame.mixer.Sound('audio/molotovbr.wav')
151 MOLOTOVBR.set_volume(3)
152
153 PICK = pygame.mixer.Sound('audio/grenadepick.mp3')
154 PICK.set_volume(2)
155
156 PICKHEALTH = pygame.mixer.Sound('audio/pills.mp3')
157 PICKHEALTH.set_volume(2)
158
159 GRUNTING = pygame.mixer.Sound('audio/Grunting.mp3')
160 GRUNTING.set_volume(2)
161
162 ZOMBIEATTACK = pygame.mixer.Sound('audio/zombieattack.mp3')
163 ZOMBIEATTACK.set_volume(20)
164
165 MENUSELECT = pygame.mixer.Sound('audio/menuselect.mp3')
166 MENUSELECT.set_volume(9)
167

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
FileNotFoundError: No file 'img/Background/pine1.png' found in working directory 'C:\Users\hp\Desktop\Zombioo-main\Zombioo-main'
PS C:\Users\hp\Desktop\Zombioo-main\Zombioo-main> & C:/ProgramData/Anaconda3/python.exe c:/Users/hp/pygame 2.1.3 (SDL 2.0.22, Python 3.9.13)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Then we have inserted respective colors and sound in the game according to our theme which attracts the viewers.



```
MAINGAME.py X game5.py game1.py game2.py
MAINGAME.py > ...
210
211 # LEVEL RESET
212 def reset_level():
213     enemy_group.empty()
214     bullet_group.empty()
215     zombiebullet_group.empty()
216     grenade_group.empty()
217     molotov_group.empty()
218     explosion_group.empty()
219     moloexplosion_group.empty()
220     item_box_group.empty()
221     decoration_group.empty()
222     water_group.empty()
223     exit_group.empty()
224
225 # TILE2
226 data = []
227 for row in range(ROWS):
228     r = [-1] * COLS
229     data.append(r)
230
231 return data
232
233
234 class Soldier(pygame.sprite.Sprite):
235     def __init__(self, char_type, x, y, scale, speed, ammo, grenades, molotovs):
236         pygame.sprite.Sprite.__init__(self)
237         self.alive = True
238         self.char_type = char_type
239         self.speed = speed
240         self.ammo = ammo
241         self.start_ammo = ammo
242         self.shoot_cooldown = 0
243         self.grenades = grenades
244         self.molotovs = molotovs
245         self.health = 130
246         self.max_health = self.health
247         self.direction = 1
248         self.vel_y = 0
249         self.jump = False
250         self.in_air = True
251         self.flip = True
```

We have made 4 levels in total with different purposes so after reaching the final level and defeating the zombie the player wins the game.

```
MAINGAME.py X game5.py game1.py game2.py
MAINGAME.py > ...
907
908 # MENU
909 screen.blit(menubg, (0, 0)) #screen.blit(menu_halloween, (0, 0)) #halloween_update
910 draw_text('DEAD RACE', DEAD_RACE, WHITE, 180, 125)
911 draw_text('@2023', KTEAM, WHITE, 905, 845)
912 draw_text('Current build V2.0 (released.21.2.2023)', ver, WHITE, 5, 845)
913
914 if start_button.draw(screen):
915     start_game = True
916     MENUSELECT.play()
917     pygame.mixer.music.stop()
918 if exit_button.draw(screen):
919     MENUSELECT.play()
920     run = False
921     pygame.display.update()
922 if settings_button.draw(screen):
923     MENUSELECT.play()
924     game_state = "settings"
925 elif game_state == "settings":
926     settings_show()
927     info_show()
928 if jk_button.draw(screen):
929     MENUSELECT.play()
930     webbrowser.open()
931 if speaker.draw(screen):
932     MENUSELECT.play()
933     speaker_sp = "speaker_menu"
934 elif speaker_sp == "speaker_menu":
935     MENUSELECT.play()
936     speaker_show()
937 if info_button.draw(screen):
938     MENUSELECT.play()
939     webbrowser.open()
940     webbrowser.open()
941 else:
942     draw_bg()
943     world.draw()
944     health_bar.draw(player.health)
945     draw_text('HEALTH', font, WHITE, 46, 12)
946     draw_text('AMMO: ', font, WHITE, 10, 35)
947     for x in range(player.ammo):
948         screen.blit(bullet_img, (90 + (x * 10), 40))
```

We have menu option to start the game or to exit from it.



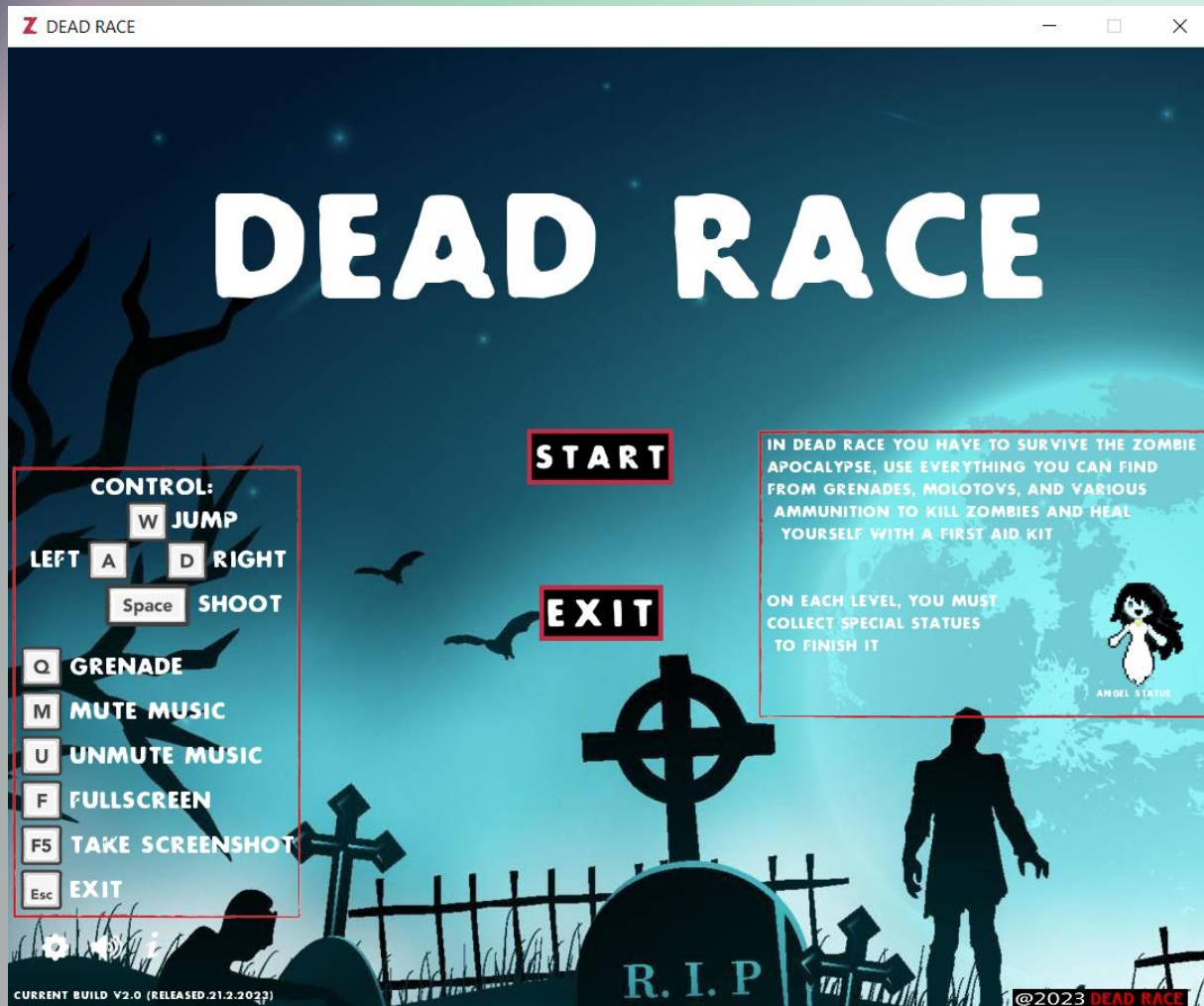

```
MAINGAME.py X game5.py game1.py game2.py
MAINGAME.py > ...
1008 # LEVEL COMPLETE
1009 if level_complete:
1010     level += 1
1011     bg_scroll = 0
1012     world_data = reset_level()
1013     if level <= MAX_LEVELS:
1014         # CREATING WORLD
1015         with open(f'level/level{level}_data.csv', newline='') as csvfile:
1016             reader = csv.reader(csvfile, delimiter=',')
1017             for x, row in enumerate(reader):
1018                 for y, tile in enumerate(row):
1019                     world_data[x][y] = int(tile)
1020             world = World()
1021             player, health_bar = world.process_data(world_data)
1022     else:
1023         screen_scroll = 0
1024         draw_text('YOU DIED!', WHITE, 260, 150), GAMEOVER.stop()
1025         screen.blit(headdeadhp, (-4, 2))
1026         if exit_button_die.draw(screen):
1027             MENUSELECT.play()
1028             run = False
1029         if restart_button.draw(screen):
1030             MENUSELECT.play()
1031             bg_scroll = 0
1032             world_data = reset_level()
1033             #CREATE WORLD DATA
1034             with open(f'level/level{level}_data.csv', newline='') as csvfile:
1035                 reader = csv.reader(csvfile, delimiter=',')
1036                 for x, row in enumerate(reader):
1037                     for y, tile in enumerate(row):
1038                         world_data[x][y] = int(tile)
1039             world = World()
1040             player, health_bar = world.process_data(world_data)
1041
1042 for event in pygame.event.get():
1043     # QUIT GAME
1044     if event.type == pygame.QUIT:
1045         MENUSELECT.play()
1046         run = False
1047     # KEYBOARD SETT
1048     if event.type == pygame.KEYDOWN:
```

To complete level you have to defeat zombies and survive, you can also quit game and restart again.

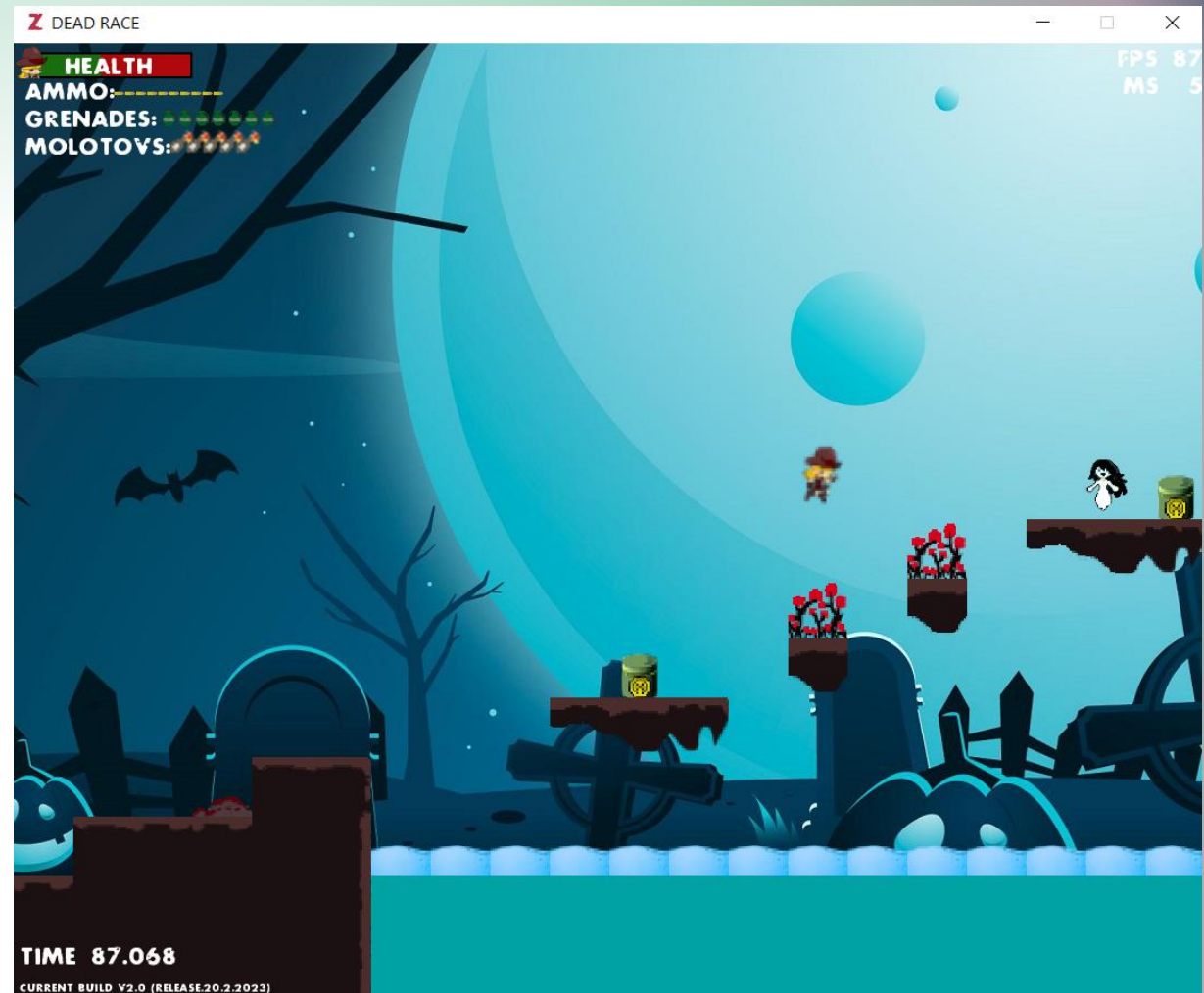
```
MAINGAME.py X game5.py game1.py game2.py
MAINGAME.py > ...
1042 for event in pygame.event.get():
1043     # QUIT GAME
1044     if event.type == pygame.QUIT:
1045         MENUSELECT.play()
1046         run = False
1047     # KEYBOARD SETT
1048     if event.type == pygame.KEYDOWN:
1049         if event.key == pygame.K_a:
1050             moving_left = True
1051         if event.key == pygame.K_d:
1052             moving_right = True
1053         if event.key == pygame.K_SPACE:
1054             shoot = True
1055         if event.key == pygame.K_q:
1056             grenade = True
1057         if event.key == pygame.K_e:
1058             molotov = True
1059         if event.key == pygame.K_w and player.alive:
1060             player.jump = True
1061         if event.key == pygame.K_ESCAPE:
1062             run = False
1063             MENUSELECT.play()
1064         if event.key == pygame.K_m:
1065             pygame.mixer.music.pause()
1066             MENUSELECT.play()
1067         if event.key == pygame.K_u:
1068             pygame.mixer.music.unpause()
1069             MENUSELECT.play()
1070         if event.key == pygame.K_f:
1071             SCREEN_WIDTH = 1080
1072             SCREEN_HEIGHT = 1920
1073             pygame.display.set_mode((1920, 1080), pygame.FULLSCREEN)
1074             MENUSELECT.play()
1075         if event.key == pygame.K_F5:
1076             takescreenshot(screen)
1077             MENUSELECT.play()
1078             SCREENSHOT.play()
1079
1080 # KEYBOARDS SETT2
1081 if event.type == pygame.KEYUP:
1082     if event.key == pygame.K_a:
```

we have used various keyboard sett like A,D and W to move forward ,backward and to jump respectively



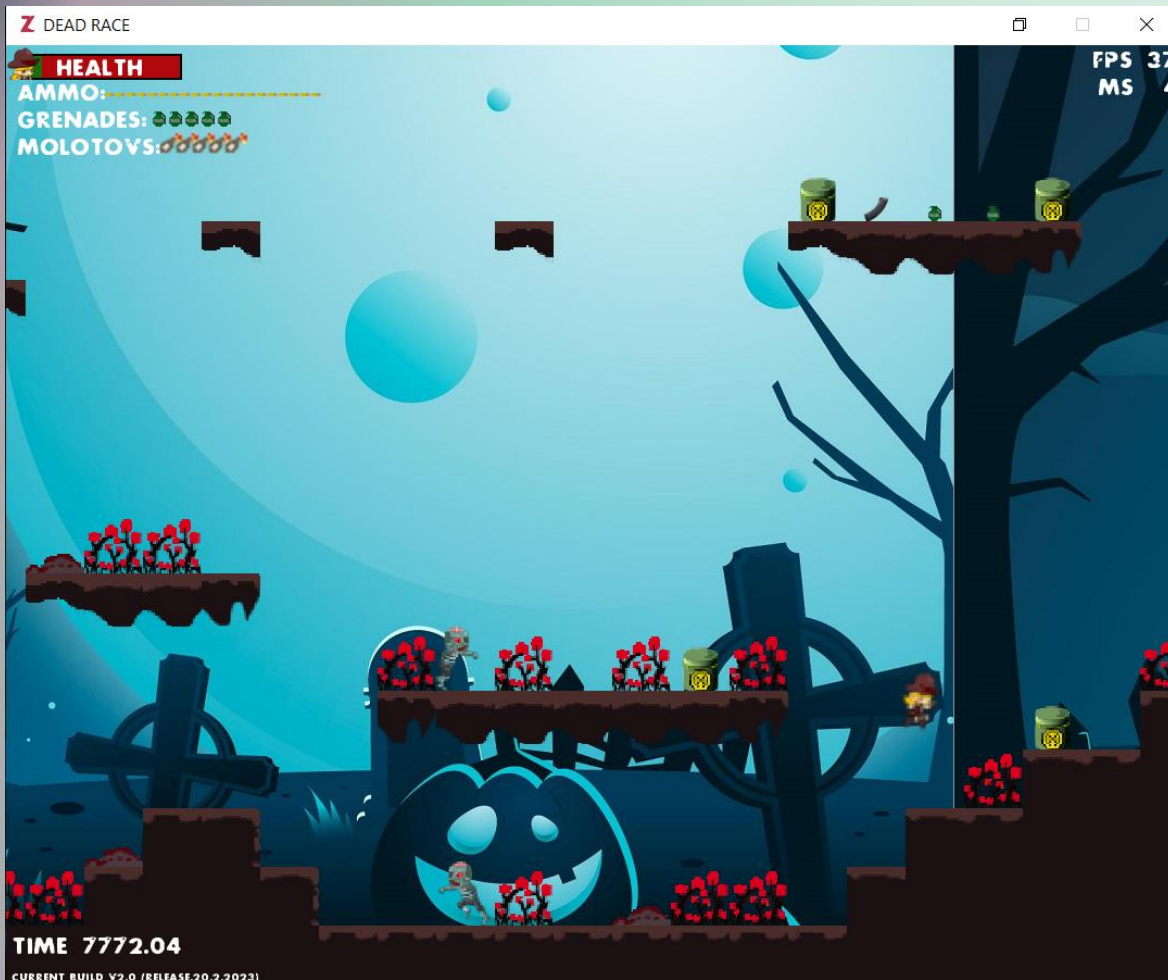


This is the front page. From here you can start the game. You can learn how to play a game, you can learn all the control keys.

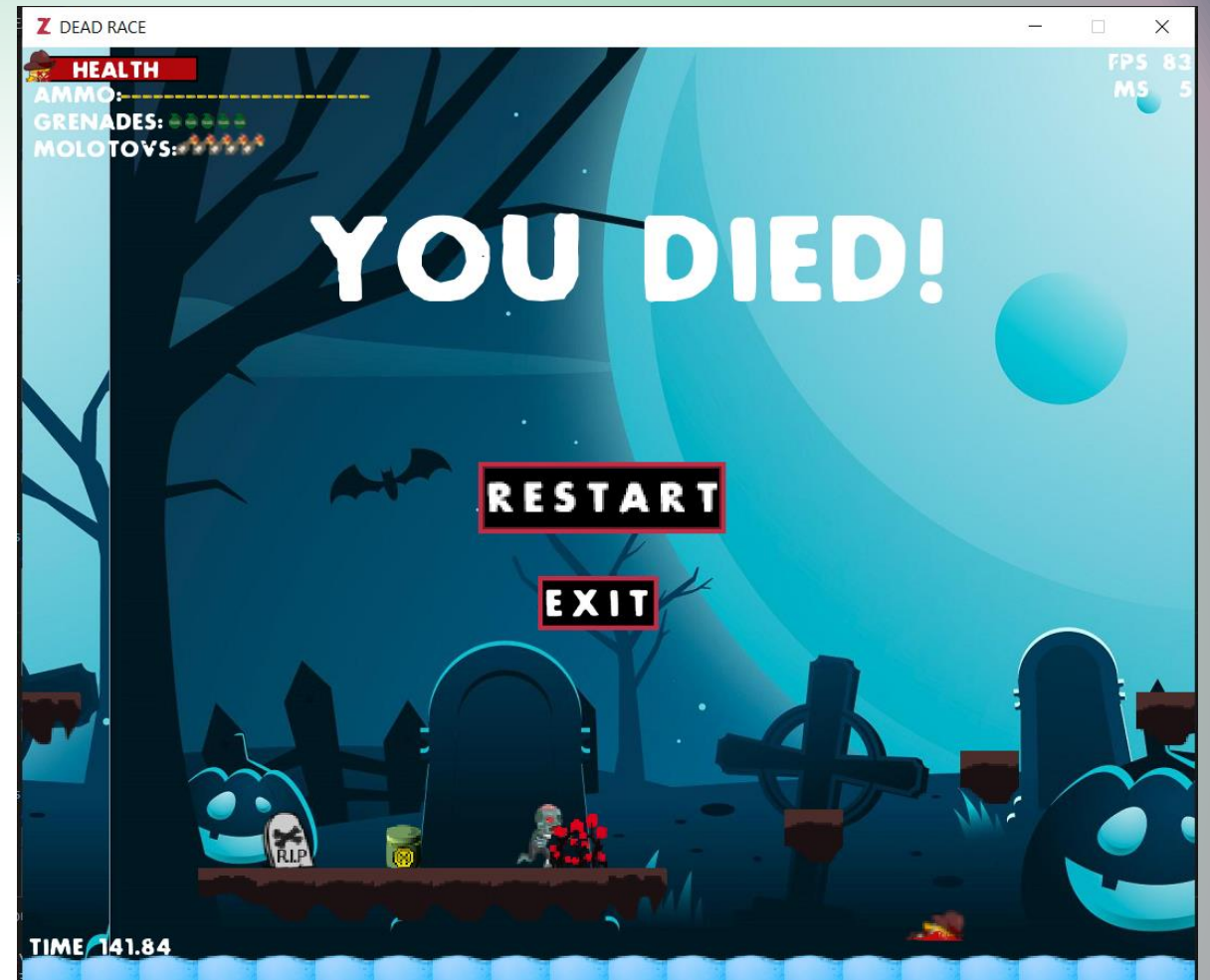


This is the player playing and crossing levels.





This is how player escapes from zombies and survive.



When zombies affect your health in this game you die, then from there you can restart the game too.





THANKS

